1. (Currently Amended) A system for interfacing applications comprising computer executable software instructions stored on a computer readable memory, said computer executable software instructions operable to implement:

a public application program interface (public API) layer useful for writing applications consistent with a set of heterogeneous vendor-specific workflow engines,

wherein each workflow engine is a software program operable to manipulate content items in accordance with a process definition,

wherein said public API layer includes a set of generic objects representing functional characteristics common to said set of heterogeneous vendor-specific workflow engines,

wherein each of said set of generic objects is a self-contained data entity, and

wherein each of said public API vendor-specific workflow engines comprises a set of generic objects an engine-specific API; and

an API adapter layer having a plurality of adapters for translating instructions from said public API layer to vendor-specific instructions,

wherein each adapter is configured to interface with a workflow engine application program interface (workflow engine API),

wherein each workflow engine API is associated with an underlying workflow engine of said set of heterogeneous vendor-specific workflow engines[[;]], and

wherein each adapter is operable to map said set of generic objects to a set of native objects for a corresponding underlying workflow engine.

2. (Original) The system of Claim 1, wherein said set of generic objects comprises a generic process definition object and wherein each of said plurality of adapters is operable to translate said generic process definition object into a native process definition object for said corresponding underlying workflow engine.

3. (Currently Amended) The system of Claim 1, said computer executable software instructions further operable to implement a content management layer, wherein said set of generic objects further comprises a payload object for indexing content items that are the subject of a process instance, wherein said process instance is an invocation of a process definition.

4.   (Currently Amended) The system of Claim 3, wherein ~~said payload object associates a set of~~ indexed content items ~~with a process instance~~ are searchable by said applications via said content management layer.

5.   (Currently Amended) The system of Claim 1, wherein ~~in~~ each of said set of ~~generic objects is~~ heterogeneous vendor-specific workflow engines maintains and operates with a vendor-specific process definition representation and wherein said applications are written to a standard process definition representation maintained ~~based upon an industry standard for workflow management~~ at said public API layer.

6.   (Currently Amended) The system of Claim [[5]] 1, wherein said ~~industry standard is~~ set of generic objects are maintained at said public API layer according to a workflow reference model promulgated by the Workflow Management Coalition.

7.   (Previously Presented) The system of Claim 6, wherein said set of generic objects comprises a WfDefinition object, a WfProcessMgr object, a WfProcess object, a WfRequestor object, a WfActivity object, a WfAssignment object, a WfResource object, a WfEventAuditBundle object, a WfUser object, a WfGroup object, and a WfRole object.

8.   (Original) The system of Claim 7, wherein said set of generic objects further comprises a WfPayload object.

9.   (Currently Amended) A system for interfacing workflow applications comprising a̶ computer executable software instructions stored on a computer readable medium, said computer executable software instructions operable to implement:

a first workflow engine;

a first workflow engine application program interface (workflow engine API) associated with said first workflow engine, wherein said first workflow engine API comprises a first set of native objects;

a second workflow engine;

a second workflow engine API associated with said second workflow engine, wherein said second workflow engine API comprises a second set of native objects, wherein said first workflow engine and said second workflow engine are heterogeneous <u>vendor-specific</u> workflow engines;

a public application program interface (public API) <u>layer</u> comprising a set of generic objects <u>representing functional characteristics common to</u> f̶o̶r̶ the heterogeneous <u>vendor-specific</u> workflow engines; <u>and</u>

<u>an API adapter layer comprising:</u>

a first adapter configured to map said set of generic objects <u>of said public API layer</u> to said first set of native objects <u>of said first workflow engine API</u>; and

a second adapter configured to map said set of generic objects <u>of said public API layer</u> to said second set of native objects <u>of said second workflow engine API</u>.

10.   (Original) The system of Claim 9, wherein said set of generic objects comprises a generic process definition object and wherein said first adapter is operable to translate said generic process definition object into to a first native process definition object and wherein said second adapter is operable to translate said generic process definition object into a second native process definition object.

11.   (Currently Amended) The system of Claim 9, <u>said computer executable software instructions further operable to implement a content management layer,</u> wherein said set of generic objects further comprises a payload object <u>for indexing content items that are the subject of a process instance, wherein said process instance is an invocation of a process definition</u>.

12. (Currently Amended) The system of Claim 11, wherein ~~said payload object associates a set of~~ indexed content items ~~with a process instance~~ are searchable by one or more applications via said content management layer.

13. (Currently Amended) The system of Claim 9, wherein each of said ~~set of generic objects is~~ first and second workflow engines maintains and operates with a vendor-specific process definition representation ~~maintained based upon an industry standard for workflow management~~.

14. (Previously Presented) The system of Claim [[13]] 9, wherein said ~~industry standard is~~ set of generic objects are maintained at said public API layer according to a workflow reference model promulgated by the Workflow Management Coalition.

15. (Previously Presented) The system of Claim 14, wherein said set of generic objects comprises a WfDefinition object, a WfProcessMgr object, a WfProcess object, a WfRequestor object, a WfActivity object, a WfAssignment object, a WfResource object, a WfEventAuditBundle object, a WfUser object, a WfGroup object, and a WfRole object.

16. (Previously Presented) The system of Claim 15, wherein said set of generic objects further comprises a WfPayload object.

17.   (Currently Amended) A computer-implemented method for integrating workflow engines comprising:

creating a public application program interface (public API) layer for at least two heterogeneous underlying workflow engines, wherein the public API layer comprises a set of generic objects,

wherein said set of generic objects represent functional characteristics common to said at least two heterogeneous underlying workflow engines,

wherein each of the at least two heterogeneous underlying workflow engines is a computer executable application program operable to manipulate content items in accordance with a process definition, and

wherein each of said at least two heterogeneous underlying workflow engines has an associated engine-specific application program interface (workflow engine API) and a set of native objects;

interfacing said public API layer with said at least two heterogeneous underlying workflow engines through said associated an API adapter layer having a plurality of adapters, wherein each workflow engine API for each of associated with said at least two heterogeneous underlying workflow engines corresponds to one of said plurality of adapters of said API adapter layer; and

mapping said set of generic objects of said public API layer to native objects of each of said at least two heterogeneous underlying workflow engines to integrate said at least two heterogeneous underlying workflow engines to said public API layer.

18.   (Currently Amended) The method of Claim 17, further comprising:

persistently maintaining a generic process definition object; and

delegating at least a portion of said set of generic objects of said public API layer to a set of corresponding native objects at one or more of said underlying workflow engines.

19.   (Previously Presented) The method of Claim 18, further comprising translating said generic process definition object into a native process definition object and persistently maintaining said native process definition object.

20.   (Currently Amended) The method of Claim 17, further comprising:

receiving a call from an application understandable by a generic object ~~from said set of generic objects~~ of said public API layer;

mapping said call from said application to a native call understandable by a native object ~~from said set of native objects~~ of a workflow engine API;

executing said native call to generate a native result; and

mapping said native result to a generic result usable by a generic object ~~from said set of generic objects~~ of said public API layer.

21.   (Currently Amended) The method of Claim ~~20~~ 17, further comprising creating a content management layer, wherein said set of generic objects further comprises a payload object for indexing content items that are the subject of a process instance, wherein said process instance is an invocation of a process definition.

22.   (Currently Amended) The method of Claim 21, wherein ~~said payload object associates a set of~~ indexed content items ~~with a process instance~~ are searchable via said content management layer.

23.   (Currently Amended) The method of Claim ~~20~~ 17, wherein each of said set of ~~generic objects is~~ heterogeneous underlying workflow engines maintains and operates with a vendor-specific process definition representation ~~maintained based upon an industry standard for workflow management~~.

24.   (Currently Amended) The method of Claim ~~23~~ 17, wherein said ~~industry standard is~~ set of generic objects are maintained at said public API layer according to a workflow reference model promulgated by the Workflow Management Coalition.

25.   (Previously Presented) The method of Claim 24, wherein said set of generic objects comprises a WfDefinition object, a WfProcessMgr object, a WfProcess object, a WfRequestor object, a WfActivity object, a WfAssignment object, a WfResource object, a WfEventAuditBundle object, a WfUser object, a WfGroup object, and a WfRole object.

26.   (Previously Presented) The method of Claim 25, wherein said set of generic objects further comprises a WfPayload object.

27.   (Currently Amended) A computer-implemented method for providing a standardized application program interface between a plurality of software applications and a plurality of workflow engines, said method comprising:

creating and maintaining a public application program interface (public API) layer comprising a set of generic software objects;

wherein said set of generic software objects represent functional characteristics common to at least two heterogeneous workflow engines;

wherein each of said at least two heterogeneous workflow engines is a computer executable application program operable to manipulate content items in accordance with a process definition;

wherein each of said at least two heterogeneous workflow engines has an engine-specific workflow engine application program interface (workflow engine API) associated therewith;

wherein each workflow engine API comprises a set of native software objects; and

wherein workflow engine APIs of said at least two heterogeneous workflow engines are vendor-specific; and

translating and mapping said set of generic software objects of said public API layer to and from said set of native software objects of said each workflow engine API through an API adapter layer having a plurality of adapters, each of which corresponds to a workflow engine API, facilitating interoperability wherein each of said plurality of software applications is operable to generate calls understandable by said set of generic software objects of said public API layer, and wherein translated calls are understandable by said at least two heterogeneous workflow engines.